

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-252403

(43)公開日 平成5年(1993)9月28日

(51)Int.Cl.<sup>5</sup>

識別記号

庁内整理番号

F I

技術表示箇所

H 0 4 N 1/41

C 8839-5C

G 0 6 F 15/66

3 3 0 H 8420-5L

H 0 4 N 7/133

Z

11/04

A 9187-5C

審査請求 未請求 請求項の数7(全19頁)

(21)出願番号

特願平4-80454

(22)出願日

平成4年(1992)3月2日

(71)出願人 000001443

カシオ計算機株式会社

東京都新宿区西新宿2丁目6番1号

(72)発明者 渡辺 亨

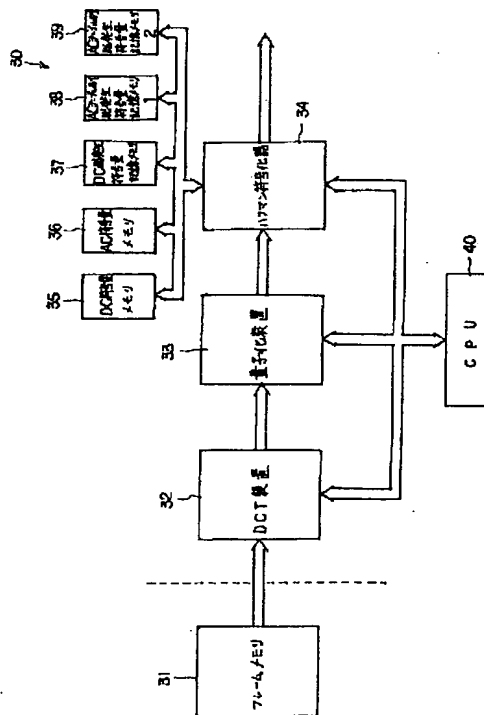
東京都羽村市栄町3丁目2番1号 カシオ  
計算機株式会社羽村技術センター内

(54)【発明の名称】 画像データ圧縮装置

(57)【要約】

【目的】 少ない処理時間で発生符号量制御を行なう。

【構成】 画像データ圧縮装置30は、Y用、C用夫々2種類の量子化テーブルを用いて量子化したブロックのAC成分によって発生した符号量の総和を記憶するACテーブル別総発生符号量記憶メモリ38、39と、量子化装置33から出力されたデータに対してハフマン符号化を施し、全てのブロックのDC成分によって発生した符号量の総和をDC総発生符号量記憶メモリ37に記憶するとともに、市松状に切換えられた2種類の量子化テーブル51～54毎のAC成分によって発生した符号量の総和をテーブル別AC総発生符号量記憶メモリ38、39に記憶するハフマン符号化器34とを設け、一度目の圧縮時に2つの量子化テーブル51と53、52と54を用いることによって量子化テーブルの変更による発生符号量の変化の予測を行ない目標とする発生する符号量に最適な量子化テーブルを作成し、二度目の圧縮によって目標とする符号量に圧縮する



## 【特許請求の範囲】

【請求項 1】 画像データを変換符号化する変換符号化手段と、

量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、

前記変換符号化手段から出力された信号を前記量子化テーブルに基づいて量子化する量子化手段と、

前記量子化手段により量子化された成分を各ブロックの予測発生符号量に従って符号化する符号化手段とを有し、

一度目の圧縮時に前記複数種の量子化テーブルを用いることにより量子化テーブルの変更による発生符号量の変化を予測し、二度目の圧縮時に目標とする符号量の圧縮を行うようにしたこと特徴とする画像データ圧縮装置。

【請求項 2】 前記変換符号化手段は、画像データに対して離散コサイン変換を実行する離散コサイン変換演算であることを特徴とする請求項 1 記載の画像データ圧縮装置。

【請求項 3】 画像データに対して離散コサイン変換を実行する離散コサイン変換演算手段と、

量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、

前記離散コサイン変換演算手段から出力された信号を、前記量子化テーブルを所定のパターンで切換えて用いて量子化する量子化手段と、

前記量子化手段により量子化された成分を事前に与えられた各ブロックの予測発生符号量に従って符号化する符号化手段と、

符号化時に各ブロックにおいて発生した DC 成分及び AC 成分の符号量を記憶する符号量記憶手段と、

前記 DC 成分及び AC 成分の符号量の総和を算出する総発生符号量算出手段と、

前記 DC 成分の総発生符号量及び所定パターンで切換えられた量子化テーブル毎の AC 成分の総発生符号量を算出する総発生符号量算出手段と、

前記総発生符号量算出手段により算出された総発生符号量を記憶する総発生符号量記憶手段とを有し、

一度目の圧縮時に前記複数種の量子化テーブルを用いることにより量子化テーブルの変更による発生符号量の変化を予測を行ない、目標とする発生する符号量に適した量子化テーブルを作成し、二度目の圧縮時に目標とする符号量の圧縮を行うようにしたこと特徴とする画像データ圧縮装置。

【請求項 4】 画像データに対して離散コサイン変換を実行する離散コサイン変換演算手段と、

量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、

前記離散コサイン変換演算手段から出力された信号を、前記量子化テーブルを市松状に切換えて用いて量子化する量子化手段と、

前記量子化手段により量子化された成分を事前に与えられた各ブロックの予測発生符号量に従って符号化する符号化手段と、

符号化時に各ブロックにおいて発生した DC 成分及び AC 成分の符号量を記憶する符号量記憶手段と、

前記 DC 成分の総発生符号量及び市松状に切換えられた量子化テーブル毎の AC 成分の総発生符号量を算出する総発生符号量算出手段と、

前記総発生符号量算出手段により算出された総発生符号量を記憶する総発生符号量記憶手段とを有し、

一度目の圧縮時に前記複数種の量子化テーブルを市松状に切換えることにより目標とする符号量に適した量子化テーブルを作成して各ブロックの発生符号量を予測し、二度目の圧縮時に目標とする符号量に圧縮するようにしたこと特徴とする画像データ圧縮装置。

【請求項 5】 画像データに対して離散コサイン変換を実行する離散コサイン変換演算手段と、

量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、

前記離散コサイン変換演算手段から出力された信号を、前記量子化テーブルをフィールド毎に切換えて用いて量子化する量子化手段と、

前記量子化手段により量子化された成分を事前に与えられた各ブロックの予測発生符号量に従って符号化する符号化手段と、

符号化時に各ブロックにおいて発生した DC 成分及び AC 成分の符号量を記憶する符号量記憶手段と、

前記 DC 成分の総発生符号量及びフィールド毎の AC 成分の総発生符号量を算出する総発生符号量算出手段と、

前記総発生符号量算出手段により算出された総発生符号量を記憶する総発生符号量記憶手段とを有し、

一度目の圧縮時には 1 つのフレームを複数のフィールドに分割し、該分割したフィールド毎に異なる量子化テーブルを用いて発生符号量を求めるとともに、該発生符号量に基づいて量子化テーブルの変更による全体及び各ブロックの発生符号量の予測を行ない、二度目の圧縮時にはフレームのデータとして目標とする符号量に圧縮するようにしたこと特徴とする画像データ圧縮装置。

【請求項 6】 前記フレームは、奇数フィールドと偶数フィールドの 2 つのフィールドに分割するようにしたこと特徴とする請求項 5 に記載の画像データ圧縮装置。

【請求項 7】 前記符号化手段は、確率分布が与えられた情報源に対してコンパクト符号を構成するハフマン符号化を行うことを特徴とする請求項 1、請求項 3、請求項 4 又は請求項 5 の何れかに記載の画像データ圧縮装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、画像データの圧縮処理等に用いられるデータ圧縮装置に係り、詳細には、発生

符号量を制御する発生符号量制御機能を有する画像データ圧縮装置に関する。

#### 【0002】

【従来の技術】画像圧縮の国際標準としてJ P E G (Joint Photographic Expert Group) やM P E G (Moving Picture Expert Group) がある。J P E Gは、静止画像を圧縮することを目的としており、すでにカラー静止画像の符号化手法が決定し、国際標準規格として承認される予定である。J P E Gについては、チップも製品化されており、このチップを用いたボードも市場に出始めている。J P E Gアルゴリズムは、大きく2つの圧縮方式に分けられる。第1の方式はD C T (Discrete Cosine Transform: 離散コサイン変換) を基本とした方式であり、第2の方式は2次元空間でD P C M (Differential PCM) を行なうSpatial (空間関数) 方式である。D C T方式は量子化を含むため一般には完全に元の画像は再現されない非可逆符号化であるが、少ないビット数においても十分な復号画像品質を得ることができ、本アルゴリズムの基本となる方式である。一方、Spatial方式は、圧縮率は小さいが元の画像を完全に再現する可逆符号化であり、この特性を実現するために標準方式として付加された方式である。

【0003】D C T方式はさらに必須機能であるベースライン・プロセス (Baseline System) とオプション機能である拡張D C Tプロセス (Extended System) の2つに分類される。これらの方式と別に、上記の方式を組み合わせてプログレッシブ・ビルドアップを実現するハイパーキカル・プロセスがある。ベースライン・プロセスは、D C T方式を実現するすべての符号器/復号器がもたなければならない最小限の機能で、A D C T方式 (Adaptive Discrete Cosine Transform Coding: 適応型離散コサイン変換) を基礎としたアルゴリズムである。上記ベースライン・プロセスにおける画像圧縮では画像データを8×8ピクセル単位のブロックで処理をする。処理プロセスは、以下の通りである。

- (1) 2次元D C T変換処理
- (2) D C T係数の量子化処理
- (3) エントロピー符号化処理

2次元D C T変換処理では、空間データを周波数データに変換し、64個のD C T係数を出力する。このとき、色成分は、(Y, CB, CR)としている。この係数のうち行列の中の左上の係数はD C成分と呼ばれ、ブロック・データの平均値である。また、残りの63個の係数は、A C成分と呼ばれる。

【0004】D C成分の量子化処理では、量子化器で各係数ごとに大きさの異なった量子化ステップ・サイズを設定した量子化マトリクスを用いて、D C T係数を線形量子化する。但し、符号量あるいは復号画品質を制御可能とするために、外部から指定する係数 (スケーリング・ファクタ) を量子化マトリクスに乗じた値を実際のマ

トリクス値として使用し、量子化を行なう。このように、テーブルを参照しながら64個のD C T係数を整数値に量子化する。この量子化処理によって非可逆圧縮となる。また、使用される参照テーブルの内容についてはJ P E Gでは規定していない。量子化のテーブルは、人間の視覚特性を考慮して作成する。人間は、高周波数成分の視覚情報には鈍いので、この高周波成分は粗く量子化する。

【0005】エントロピー符号化処理では、まずD C成分と左隣ブロックにおける量子化されたD C成分との差分を計算し、符号化する。この方法は、D P C Mと呼ばれる。また、A C成分はジグザグ・スキャンにより1次元配列に変換される。ベースラインプロセスのエントロピー符号化では、ハフマン符号化方式を用いる。ハフマン符号化処理では各係数がゼロであるかどうかを判定し、連続するゼロの係数は、その長さがランレングスとして勘定される。ゼロでない係数が来ると、その量子化結果とそれまでのゼロ係数のランレングスを組み合わせて、2次元ハフマン符号化される。D C/A C係数のハフマン符号化は、与えられたハフマン符号テーブルに基づくが、量子化マトリクスおよびハフマン符号テーブルは、使用する状況において最適なものになるようにするためデフォルト値はなく、必要に応じて符号器から復号器へ転送して使用する。

【0006】図13は上記J P E Gアルゴリズムに基づく従来の画像データ圧縮装置10のブロック図である。図13において、11はフレームメモリであり、原画像のデータを保持してある。フレームメモリ11に蓄えられている画像データは、D C T演算部12に出力され、データを読み込んでブロック単位のD C T演算を行ない量子化部13に出力する。D C T演算の出力データは、量子化装置13 (図14) に読込まれ、量子化装置13は、量子化テーブルRAM14の量子化テーブル (図15及び図16参照) の値に従って量子化演算を行なう。ハフマン符号化部14は、量子化装置13から出力されたデータに対してD C符号量メモリ15及びA C符号量メモリ16を用いてハフマン符号化を施し、総発生符号量を総発生符号量記憶メモリ17に記憶する。C P U 18は、これら装置全体の制御、及び後述する最適な量子化テーブル作成と各ブロックの発生符号量を予測する制御を行なう。上記ハフマン符号化器14は、量子化装置13から出力されたデータに対してハフマン符号化を行なう。このハフマン符号化器14の動作には、以下のような2種類のモードmode 1, mode 2がある。

(mode 1) 入力データに対してハフマン符号化を行ない、その際に、各ブロックのD C成分によって発生したビット数を、D C符号量メモリ15に書込む。また、A C成分によって発生したビット数をA C符号量メモリ16に書込む。

(mode 2) 入力データに対してハフマン符号化を行

ない、その際に事前にAC符号量メモリ16に書き込まれている各ブロックのDC成分とAC成分の両方による発生符号量の予測値と実際に発生した符号量とを比較し、もし発生符号量が予測よりも多く発生するのであれば、そこである値以降は全て0とするEOB (End of Block) を発生させ、そのブロックの符号化を終了する。また、余ったビット数は次ブロックの予測値に加えられる。

【0007】図14は上記量子化装置13のブロック図である。図14において、量子化装置13は、Y（輝度情報）用量子化テーブル（図15）を記憶する量子化テーブルRAM21と、C（色情報）用量子化テーブル

（図16）を記憶する量子化テーブルRAM22と、DCT演算された各周波数成分に量子化テーブルの値を掛ける（すなわち、逆数を掛けて除算する）ことによって重み付けを変える量子化を行なう乗算器23とにより構成されている。また、量子化テーブルRAM21、22は、Y及びR-Y、B-Yの各コンポーネントそれぞれに対応している。

【0008】図15はY用量子化テーブル、図16はC用量子化テーブルを示す図であり、上記量子化テーブルRAM21、22内には、実際には、量子化テーブルの逆数が書込まれている。量子化装置13は、この乗算結果を出力する。以上の構成において、上記画像データ圧縮装置10は以下のような動作により最適な量子化テーブルと各ブロックにおける発生符号量の予測を行なう。

i) 先ず、適当な量子化テーブルを作成し、量子化テーブルRAM21、22に書込む。

ii) 次いで、DCT→量子化→ハフマン符号化 (mode 1) を行なう。

iii) AC符号量メモリ16に保持されている総発生符号量を参照し、その値を基に新たに適当な量子化テーブルを作成し、作成した量子化テーブルを量子化テーブルRAM21、22に書込む。

iv) 次いで、DCT→量子化→ハフマン符号化 (mode 1) を行なう。

v) 上記ii) で得られた総発生符号量と、今回得られた総発生符号量から目標とする符号量を発生させる量子化テーブルを作成し、量子化テーブルRAM21、22に書込む。また、目標とする符号量からDC成分によって発生した総符号量及び各ブロックのEOB長を減算し、減算した値とAC成分によって発生した総符号量との比率を求める。そして、上記比率に各ブロックのAC成分によって発生した符号量からEOB長を減算した値を乗じ、その値に各ブロックのDC成分によって発生した符号量にEOB長を加えた値を各ブロックの発生符号量の予測値としてAC符号量メモリ15に書込む。

vi) そして、DCT→量子化→ハフマン符号化 (mode 2) を行なう。

【0009】

【発明が解決しようとする課題】しかしながら、このような従来の画像データ圧縮装置10にあっては、上述した方法によって符号量制御を行っていたため、上記i)、ii)、iv)及びvi)に示す“DCT→量子化→ハフマン符号化”というシーケンスを3度も実行する必要がある、そのため処理に多くの時間を必要とするという欠点があった。そこで本発明は、少ない処理時間で発生符号量制御を行うことが可能な画像データ圧縮装置を提供することを目的とする。

【0010】

【課題を解決するための手段】請求項1記載の発明は、上記目的達成のため、画像データを変換符号化する変換符号化手段と、量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、前記変換符号化手段から出力された信号を前記量子化テーブルに基づいて量子化する量子化手段と、前記量子化手段により量子化された成分を各ブロックの予測発生符号量に従って符号化する符号化手段とを有し、一度目の圧縮時に前記複数種の量子化テーブルを用いることにより量子化テーブルの変更による発生符号量の変化を予測し、二度目の圧縮時に目標とする符号量の圧縮を行うようにしている。前記変換符号化手段は、例えば請求項2に記載されているように画像データに対して離散コサイン変換を実行する離散コサイン変換演算であつてもよい。請求項3記載の発明は、画像データに対して離散コサイン変換を実行する離散コサイン変換演算手段と、量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、前記離散コサイン変換演算手段から出力された信号を、前記量子化テーブルを所定のパターンで切換えて用いて量子化する量子化手段と、前記量子化手段により量子化された成分を事前に与えられた各ブロックの予測発生符号量に従って符号化する符号化手段と、符号化時に各ブロックにおいて発生したDC成分及びAC成分の符号量を記憶する符号量記憶手段と、前記DC成分及びAC成分の符号量の総和を算出する総発生符号量算出手段と、前記DC成分の総発生符号量及び所定パターンで切換えられた量子化テーブル毎のAC成分の総発生符号量を算出する総発生符号量算出手段と、前記総発生符号量算出手段により算出された総発生符号量を記憶する総発生符号量記憶手段とを有し、一度目の圧縮時に前記複数種の量子化テーブルを用いることにより量子化テーブルの変更による発生符号量の変化を予測を行ない、目標とする発生する符号量に適した量子化テーブルを作成し、二度目の圧縮時に目標とする符号量の圧縮を行うようにしている。請求項4記載の発明は、画像データに対して離散コサイン変換を実行する離散コサイン変換演算手段と、量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、前記離散コサイン変換演算手段から出力された信号を、前記量子化テーブルを市松状に切換えて用いて量子化する量子化手

段と、前記量子化手段により量子化された成分を事前に与えられた各ブロックの予測発生符号量に従って符号化する符号化手段と、符号化時に各ブロックにおいて発生したDC成分及びAC成分の符号量を記憶する符号量記憶手段と、前記DC成分の総発生符号量及び市松状に切換えられた量子化テーブル毎のAC成分の総発生符号量を算出する総発生符号量算出手段と、前記総発生符号量算出手段により算出された総発生符号量を記憶する総発生符号量記憶手段とを有し、一度目の圧縮時に前記複数種の量子化テーブルを市松状に切換えることにより目標とする符号量に適した量子化テーブルを作成して各ブロックの発生符号量を予測し、二度目の圧縮時に目標とする符号量に圧縮するようにしている。請求項5記載の発明は、画像データに対して離散コサイン変換を実行する離散コサイン変換演算手段と、量子化演算に用いる複数種の量子化テーブルを記憶する量子化テーブル記憶手段と、前記離散コサイン変換演算手段から出力された信号を、前記量子化テーブルをフィールド毎に切換えて用いて量子化する量子化手段と、前記量子化手段により量子化された成分を事前に与えられた各ブロックの予測発生符号量に従って符号化する符号化手段と、符号化時に各ブロックにおいて発生したDC成分及びAC成分の符号量を記憶する符号量記憶手段と、前記DC成分の総発生符号量及びフィールド毎のAC成分の総発生符号量を算出する総発生符号量算出手段と、前記総発生符号量算出手段により算出された総発生符号量を記憶する総発生符号量記憶手段とを有し、一度目の圧縮時には1つのフレームを複数のフィールドに分割し、該分割したフィールド毎に異なる量子化テーブルを用いて発生符号量を求めるとともに、該発生符号量に基づいて量子化テーブルの変更による全体及び各ブロックの発生符号量の予測を行ない、二度目の圧縮時にはフレームのデータとして目標とする符号量に圧縮するようにしている。前記フレームは、例えば請求項6に記載されているように奇数フィールドと偶数フィールドの2つのフィールドに分割するようにしたものであってもよく、また、前記符号化手段は、例えば請求項7に記載されているように確率分布が与えられた情報源に対してコンパクト符号を構成するハフマン符号化を行なうものであってもよい。

【0011】

【作用】本発明の手段の作用は次の通りである。請求項1及び請求項2記載の発明では、画像データは離散コサイン変換演算手段で離散コサイン変換された後、量子化手段に送られる。量子化手段では、DCT演算出力に対して量子化テーブルを参照して量子化演算が行われ、その演算結果が符号化手段に出力される。符号化手段では量子化データが各ブロックの予測発生符号量に従って符号化され、符号化時は各ブロックにおいて発生したDC成分及びAC成分の符号量が符号量記憶手段に記憶される。また、総発生符号量算出手段によってDC成分及び

AC成分の符号量の総和が算出され総発生符号量記憶手段に記憶される。この場合において、一度目の圧縮時には複数種の量子化テーブルが複数同時に使用されて量子化テーブルの変更による発生符号量の変化が予測され、二度目の圧縮時には目標とする符号量の圧縮が行われる。従って、符号化制御のための動作シーケンスを減少させることができ、少ない処理時間で発生符号量制御を行なうことができる。請求項3及び請求項4記載の発明では、総発生符号量算出手段により各ブロック図において発生したDC成分の総発生符号量と所定パターン（例えば、市松模様状）で切換えられた量子化テーブル毎のAC成分の総発生符号量が算出される。この場合、一度目の圧縮時には複数種の量子化テーブルが市松状に切換えられて目標とする符号量に適した量子化テーブルが作成されて各ブロックの発生符号量が予測される。そして、二度目の圧縮により目標とする符号量に圧縮される。従って、少ない処理時間で発生符号量制御を行なうことができ、処理の高速化を図ることができる。請求項5及び請求項6記載の発明では、総発生符号量算出手段によりDC成分の総発生符号量及びフィールド毎のAC成分の総発生符号量が算出される。この場合、一度目の圧縮時には1つのフレームが2つのフィールド（例えば、奇数フィールドと偶数フィールド）に分割されてフィールド毎に異なる量子化テーブルが用いられる。このようにして求めた発生符号量に基づいて量子化テーブルの変更による全体及び各ブロックの発生符号量の予測が行われる。そして、二度目の圧縮時でフレームのデータとして目標とする符号量に圧縮される。従って、2度の圧縮で正確な符号量制御を行なうことができ、フレーム制御を行なう電子カメラ等に適用することができる。

【0012】

【実施例】以下、本発明を図面に基づいて説明する。

#### 原理説明

先ず、本発明の基本的な考え方を説明する。本画像データ圧縮装置は、DCT装置等の変換符号化装置と、複数種の量子化テーブル記憶装置と、それを適当な切換えて用いる量子化装置と、事前に与えられた各ブロックの予測発生符号量に従って符号化（例えば、ハフマン符号化）を行なう符号化器と、各ブロックにおいて発生したDC成分の符号量及びAC成分の符号量を記憶する記憶装置と、DC成分の総発生符号量及び適当に切換えられた量子化テーブル毎のAC成分の総発生符号量を算出する算出装置と、算出した総発生符号量を記憶する記憶装置等を有し、一度目の圧縮時に複数の量子化テーブルを用いることによって量子化テーブルの変更による発生符号量の変化の予測を行ない、目標とする発生符号量に最適な量子化テーブルを作成し、二度目の圧縮によって目標とする符号量に圧縮するようにするものである。すなわち、本画像データ圧縮装置では発生符号量制御において、一度目のパスのときに2つの量子化テーブルを使っ

てしまうことによって従来3度行っていた“DCT→量子化→符号化”動作を2度に減らすようにするものである。以下、一度目の量子化において複数種の量子化テーブルを用いる具体的方法については第1及び第2の実施例で詳述する。

#### 【0013】第1実施例

図1～図9は本発明に係る画像データ圧縮装置の第1実施例を示す図であり、本実施例は複数種の量子化テーブルを用いる方法として各ブロックに市松状に2種類の量子化テーブルを使用するようにした例である。先ず、構成を説明する。図1は画像データ圧縮装置30のブロック図であり、この図において、画像データ圧縮装置30は、データ圧縮すべき原画像データを記憶するフレームメモリ31と、フレームメモリ31から読出した画像データを2次元DCTを施してDCT係数を求めるDCT演算を行なうDCT装置32と、DCT演算の出力データを量子化テーブルRAM（後述）の値に従って量子化演算する量子化装置33と、後述する各メモリ35～39を用いて従来例と同様の2つの動作モードmode 1, mode 2に従って量子化装置33の出力に対してハフマン符号化を行なうハフマン符号化器34と、DC成分によって発生したDC符号量を記憶するDC符号量メモリ35と、AC成分によって発生したAC符号量を記憶するAC符号量メモリ36と、全てのブロックのDC成分によって発生した符号量の総和を記憶するDC総発生符号量記憶メモリ37と、Y用、C用夫々2種類の量子化テーブルを用いて量子化したブロックのAC成分によって発生した符号量の総和を記憶するACテーブル別総発生符号量記憶メモリ38、39と、これら全体の制御及び最適と思われる量子化テーブル及び、その時の各ブロックの発生符号量の予測値の算出を行なうCPU40とにより構成されている。

【0014】上記ハフマン符号化器34は、量子化装置33から出力されたデータに対してDC符号量メモリ35及びAC符号量メモリ36を用いてハフマン符号化を施し、全てのブロックのDC成分によって発生した符号量の総和をDC総発生符号量記憶メモリ37に記憶するとともに、市松状（図3及び図4参照）に切り換えられた後述する2種類の量子化テーブル51～54毎のAC成分によって発生した符号量の総和をテーブル別AC総発生符号量記憶メモリ38、39に記憶する。

【0015】また、上記テーブル別AC総発生符号量記憶メモリ38、39のうち、AC総発生符号量記憶メモリ38には、量子化テーブル51、52を、またAC総発生符号量記憶メモリ39には、量子化テーブル53、54を用いて量子化したブロックのAC成分によって発生した符号量の総和が記憶保持される。また、後述する（mode b）の時には、AC総発生符号量記憶メモリ38にのみ値が書込まれ、AC総発生符号量記憶メモリ39は、0である。

【0016】図2は上記量子化装置33のブロック図である。図2において、量子化装置33は、C用の各ブロックに対して市松状としたときの市松模様の一方向のブロックにおけるY用量子化テーブル51を記憶する量子化テーブルRAM41と、C用の各ブロックに対して市松状としたときの市松模様の他方向のブロックにおけるY用量子化テーブル53を記憶する量子化テーブルRAM43と、R-Y、B-Y用の各ブロックに対して市松状としたときの市松模様の一方向のブロックにおけるR-Y、B-Y用量子化テーブル52を記憶する量子化テーブルRAM42と、R-Y、B-Y用の各ブロックに対して市松状としたときの市松模様の他方向のブロックにおけるR-Y、B-Y用量子化テーブル54を記憶する量子化テーブルRAM44と、DCT演算の結果と上記量子化テーブル51～54の値を乗算する（逆数を掛けて除算することによって重み付けを変える量子化を行なう乗算器45とにより構成されている。

【0017】図3は量子化テーブルRAM41、43に記憶されたY用量子化テーブル51、53、図54は量子化テーブルRAM42、54に記憶された量子化テーブル52、54を示す図である。図3に示すようにY用量子化テーブル51、53は、各ブロックに対して市松状にY用量子化テーブル51とY用量子化テーブル53（図3ハッチング部参照）とにより構成される。すなわち、従来例では図15に示すように各ブロックが全体として1つのY用量子化テーブルを構成していたが、本Y用量子化テーブル51、53は各ブロックを図3のような市松模様状とし、図3ハッチング部に示すブロックには量子化テーブル53を、非ハッチング部に示すブロックには量子化テーブル51を割り当てるようにする。同様に、R-Y、B-Y用量子化テーブル52、54は、各ブロックに対して市松状にR-Y、B-Y用量子化テーブル52とR-Y、B-Y用量子化テーブル54（図4ハッチング部参照）とにより構成され、各ブロックを図4のような市松模様状とし、図4ハッチング部に示すブロックには量子化テーブル54を、非ハッチング部に示すブロックには量子化テーブル52を割り当てるようにする。ここで、実際には市松模様にするると割り当てられるブロック数が半分になるので算出される発生符号量も約半分になる。そこで、本実施例では算出された発生符号量を2倍して使うようにする。このように、量子化テーブル51と53、52と54を切り換えることによって2つのテーブルに基づく2点の発生符号量の傾向（リニアの場合にあってはその傾き）が求まるので、算出された発生符号量に対応するスケーリング値を1回目の量子化で求めることができるようになる。

【0018】また、テーブル1として用いられるY用量子化テーブル51及びR-Y、B-Y用量子化テーブル52と、テーブル2として用いられるY用量子化テーブル53及びR-Y、B-Y用量子化テーブル54は、例

えば図5に示すような値のものが使用される。

【0019】上記量子化テーブルRAM41～44に記憶される量子化テーブル51～54はコンポーネント及びブロックに対して以下のように切換えて用いられる。

(mode a) Yに対してはY用量子化テーブル51, 53を用いる。また、このY用量子化テーブル51, 53を図4に示すような市松状に各ブロック毎に切換える。R-Y, B-Yについては、R-Y, B-Y用量子化テーブル52, 54を用いる。切換え方法はYと同様に市松状に切換える。

(mode b) Yに対してY用量子化テーブル51を、R-Y, B-Yに対してR-Y, B-Y用量子化テーブル52を用いる。この場合、ブロック毎の切換えは行われない。

【0020】次に、本実施例の動作を説明する。

#### 全体動作

まず、フレームメモリ31には、原画像のデータが保持されているものとする。DCT演算部32はフレームメモリ31から画像データを読み込んでDCT演算を行ない、そのDCT演算出力は、量子化装置33に送られる。また、量子化装置33では、DCT演算装置32からの入力に対して量子化テーブルRAM41～44の値に従って量子化演算を行ない、演算結果をハフマン符号化器34に出力する。ハフマン符号化器34は、量子化されたデータに対してDC符号量メモリ35及びAC符号量メモリ36を用いてハフマン符号化を施し、全てのブロックのDC成分によって発生した符号量の総和をDC総発生符号量記憶メモリ37に記憶するとともに、市松状に切換えられた2種類の量子化テーブル51～54毎のAC成分によって発生した符号量の総和をテーブル別AC総発生符号量記憶メモリ38, 39に書き込む。この場合、Y用量子化テーブル51(図3の非ハッチング部参照)を使ったブロックの発生符号量の総和はACテーブル別総発生符号量記憶メモリ38に書き込まれ、Y用量子化テーブル51(図3のハッチング部参照)を使ったブロックの発生符号量の総和はACテーブル別総発生符号量記憶メモリ39に書き込まれる。

\*

$$S_b + \frac{(CDN - DC_{all} - A(s_b \times 2) \times (S_h - S_b))}{(AC_{sh} - AC_{sb}) \times 2}$$

【0023】次に、以下のようなアルゴリズムによって各ブロックの発生符号量の予測を行なう。図6～図8は各ブロックの発生符号量の予測処理を示すフローチャートであり、図6～図8のフローチャートで使用される変数の対応を以下に示す。

CDN: 目標とするbit数

RN: 未配分のbit数

RAT[0], RAT[1]: 目標とするbit数におけるAC成分の発生符号量と、量子化テーブル51, 52及び53, 54を用いた時の総発生符号量の2倍との比率予測値の1/6倍

\*【0021】CPU40は、上記各装置を以下のように動作させ、発生符号量の制御を行なう。

i) 先ず、目標とする総発生符号量に合わせて量子化テーブルを作成し、量子化テーブル51, 52, 53, 54にその逆数を書込む。図5は1bppの場合の量子化テーブル51～54の一例である。

ii) DCT→量子化(mode a)→ハフマン符号化(mode 1)を行なう。

iii) 後述のアルゴリズムに従って目標とする符号量を発生させるのに最適と思われる量子化テーブルと、その時の各ブロックのAC成分及びDC成分の発生符号量を予測し、その値を量子化テーブル51, 52、DC符号量メモリ35及びAC符号量メモリ36に書き込む。

iv) 次に、DCT→量子化(mode b)→ハフマン符号化(mode 2)を行なう。

#### 【0022】発生符号量予測アルゴリズム

先ず、量子化テーブル作成アルゴリズムについて述べる。

①事前に発生符号量がリニアとなるような量子化テーブルの作成方法を決めて番号付けをしておく。この番号をスケージングと呼ぶことにする。例えば、0～255の256通りとし、0から255へ向かってリニアに発生符号量が増加するものとする。

②目標とする発生符号量(CDN)に対して適当なスケージングを2つ選択し、量子化テーブルを作成し、量子化テーブル51, 52, 53, 54にその逆数を書込み、量子化(mode a)及びハフマン符号化(mode 1)を行なう。図5は、このようにして作成される量子化テーブル例である。

③最初に選択したスケージング値をS<sub>b</sub>, S<sub>h</sub>とする。また、DC総発生符号量記憶メモリ37のDC総発生符号量の値をDC<sub>all</sub>、上記S<sub>b</sub>, S<sub>h</sub>に対応するACテーブル別総発生符号量記憶メモリ38, 39のテーブル別AC総発生符号量の値をそれぞれAC<sub>sb</sub>, AC<sub>sh</sub>とすると、求めるスケージングは、例えば数1によって得ることができる。

【数1】

EOB[0], EOB[1]: 0: YのEOB長  
1: CのEOB長

AC[i][j][k]: i→0/1=Y/C

j=k, j行k列のブロックのAC発生符号量

DC[i][j][k]: i→0/1=Y/C

j=k, j行k列のブロックのDC発生符号量

F[i][j][k]: i→0/1=Y/C

j=k, j行k列のブロックの発生符号量予測値

RA: 全てのブロックに分配するbit数

YA: Yのブロックに分配するbit数

RB: 全ての左端のブロックに分配するbit数

YB: Yの左端のブロックに分配するbit数

【0024】図6～図8のフローチャートを説明するにあたり、先ずフロー全体のアルゴリズムを簡単に述べる。

(I) 目標とする発生符号量から、一度目で発生したDC成分の符号量を減じた値と、各テーブルによって発生したAC成分の総和(ACsb, ACsh)との比率を求める。

(II) 各ブロックで実際に発生したAC発生符号量から各コンポーネントに対応するEOB長(EOB[0], EOB[1])を減じ、それに上記(I)で求めた比率を、一度に用いた量子化テーブルに従って乗じ、その値にEOBと一度目で求めたDC発生符号量を加えた値を仮の予測値とする。

(III) 上記(II)で求めた予測値の総和と目標とする発生符号量との差を求めて、各ブロックに対して以下のように割振る。

a) 余りを全体のブロック数で割り、各ブロックに均等に分配する。

b) さらに、その余りをYのブロック数で割り、Yのブロックのみに均等に分配する。

c) また、そのさらに余りをV方向のブロック数の4倍で割り、Y, R-Y, B-Yの一番左側のブロックに2:1:1の割合で分配する。

d) その余りを、Yの一番左側のブロックに均等に分配する。

e) さらにその余りは、Yの左上端のブロックに加える。

以上のようなアルゴリズムによって各ブロックの予測発生符号量を得る。

【0025】より詳しく説明すると、図6～図8において、先ず、ステップS1で未配分のビット数RNを目標とするビット数CDNとし( $RN=CDN$ )、目標とするビット数CDNから一度目で発生したDC成分の総発生符号量DCallを減じた値と各テーブルによって発生したAC成分の総和(ACsb, ACsh)との比率RAT[0], RAT[1]を算出する( $RAT[0] = (CDN - DC_{all}) \times 8 / AC_{sh}$ ,  $RAT[1] = (CDN - DC_{all}) \times 8 / AC_{ch}$ )。ここで、値( $CDN - DC_{all}$ )を $\times 8$ しているのはバイナリポイントをシフトさせて演算精度を保つためのものであり必ずしも必須のものではない。例えば、C言語で圧縮/伸長プログラムを作成した場合、C言語の除算は少数部分は切捨てとなるのでここで適当な倍数を掛けておくようにする。

【0026】次いで、ステップS2でYあるいはCのブロックを示すバンク(bank)iを0にし( $i=0$ )、ステップS3でバンクiが2より小さいか( $i < 2$ )か否かを判別する。すなわち、色(C)については横方向に2倍にサンプリングするので、バンク1とバン

ク2の2つ合わせてYと同じ大きさになる。そのため、バンク1とバンク2で配列を使ってバンクiが2になるまで以降の処理を繰返す。図9はY, B-Y, R-Yのブロックの構造を示す図であり、各ブロックはj行k列で示される。バンクiが2より小さいときはステップS4で縦方向のブロック数jを0にし( $j=0$ )、ステップS5で縦方向のブロック数jが縦方向の所定ブロック数Vより小さいか( $j < V$ )か否かを判別する。縦方向のブロック数jが縦方向の所定ブロック数V以上のときはステップS6でバンクiをインクリメント( $i=i+1$ )してステップS3に戻る。縦方向のブロック数jがVより小さいときはステップS7で横方向のブロック数kを0にし( $k=0$ )、ステップS8で横方向のブロック数kが横方向の所定ブロック数Hより小さいか( $k < H$ )か否かを判別する。横方向のブロック数kが横方向の所定ブロック数H以上のときはステップS9で縦方向のブロック数jをインクリメント( $j=j+1$ )してステップS5に戻る。例えば、縦方向のブロック数Vが60であれば、縦方向のブロック数jは0でイニシャライズされているからj=0からカウントしj=60になると、ステップS7以降で同様の判断を横方向について行なう。これによって、次のステップS10に示す処理が縦方向、横方向規定のブロック数になるまで繰返し実行され、バンクiが2になると図6のフローを抜けて図7のフローのステップS11に進む。

【0027】ステップS10では以下のような各処理を行なう。先ず、各ブロックで実際の発生したAC発生符号量AC[i][j][k]から各コンポーネントに対応するEOB長EOB[i]を減じてテンポラリtempに格納する( $temp = AC[i][j][k] - EOB[i]$ )。ここで、各ブロックで発生したAC発生符号量AC[i][j][k]からEOB[i]を減算しているのは、後の処理でオーバーフローを防止するためにEOBを加えるようにすることになるため予めこのEOB長分を減算しておくものである。また、上記テンポラリtempに所定の起立RATを乗じたものを新しいテンポラリtempの値とする( $temp = temp \times RAT[(j+k) \% 2]$ )。ここで、 $(j+k) \% 2$ とは、各ブロックにおいて市松状(図3及び図4参照)に0と1とが切替えるためのものであり、上記%2は2で割った余り(0か1のどちらかになる)を示している。これによって、市松状のどちらの比率RATを使うのが決定される。また、このテンポラリtempの値を4bit左にシフトさせる( $temp = temp \gg 4$ )。ここで、4bit左シフトするのは上記ステップS1で $\times 8$ 倍したものを元に戻すためである。また、このテンポラリtempの値に一度目で求めたDC発生符号量DC[i][j][k]とコンポーネントに対応するEOB[i]を足して新しいテンポラリtempの値とし( $temp = DC[i][j][k] + EOB[i]$ )、このテンポラリtempの値を該当ブロックの発



生符号量  $F[i][j][k]$  とする ( $F[i][j][k] = \text{temp}$ )。この  $F[i][j][k]$  が実際の予測発生符号量となる。また、未配分のビット数 (残りのビット数)  $RN$  は、それまでの未配分のビット数  $RN$  から上記求めたテンポラリ  $\text{temp}$  の値を減算したものとなる ( $RN = RN - \text{temp}$ )。そして、横方向のブロック数  $k$  をインクリメント ( $k = k + 1$ ) してステップ  $S10$  の処理を終えステップ  $S8$  に戻る。上記図 6 のフローにより各ブロックの予測発生符号量  $F[i][j][k]$  を得ることができる。

【0028】ところで、演算精度が非常に高い装置 (例えば、FPU を使用した画像データ圧縮装置) であれば図 6 のアルゴリズムだけで十分な場合もあるが、本実施例では以下図 7 及び図 8 のフローに示すようなアルゴリズムによって余り部分の割振りを行ってより精度の高い予測発生符号量を求めるようにしている。余りが生じる理由は次のような理由からである。すなわち、実際には上記比率の計算をすると完全には割り切れずどうしても余りが生じることになる。このビットの余りを四捨五入して丸めると、四捨五入による誤差で何ビットかオーバーしてしまう場合があり、結果的には全体で何ビット

(あるいは何バイト) 分もオーバーすることがある。目標値からのオーバーは許されないので、実際には四捨五入はせず、演算は基本的に切捨てで行っている。このように切捨てで行った場合、ブロック数が 1 万以上あるから、例えば 1 ブロックあたり平均 5 ビットずつ余りが出たとしても全体では 5 千ビットもの余りが発生することになる。本実施例ではこの余りを図 7 及び図 8 のフローに示すアルゴリズムで適切に分配して画質情報を大幅に向上させるようにしている。

【0029】図 7 において、ステップ  $S11$  では以下のような各処理を行なう。まず、未配分の残りビット数  $RN$  を全体のブロック数 ( $V \times H \times 2$ ) で割り全てのブロックに配分するビット数  $RA$  を求める ( $RA = RN / (V \times H \times 2)$ )。例えば、図 9 に示す  $Y$  のブロックに 100 ビット、 $B-Y$ 、 $R-Y$  のブロックに夫々 50 ビット、余りのビット数が 380 とすると、式 2 に示すように余り 380 を全体のブロック数 ( $100 + 50 + 50$ ) で割る切り捨てるの除算を行って各ブロックに例えば 1 ビットずつ分配するビット数  $RA$  を求める。

$380 / (100 + 50 + 50) = 1 = RA \dots$  (式 2)  
そして、元の未配分のビット数  $RN$  から各ブロックに分配するビット数 ( $RA \times V \times H \times 2$ ) を減算して上記均等分配で分配しきれなかった余りのビット数  $RN$  を求める ( $RN = RN - RA \times V \times H \times 2$ )。

【0030】さらに、上記均等分配で分配しきれなかった余りのビット数  $RN$  を  $Y$  のブロック数 ( $V \times H$ ) で割り、 $Y$  のブロックのみに均等に分配するビット数  $YA$  を求める ( $YA = RN / (V \times H)$ )。すなわち、 $Y$  と色

の符号を割り振ったほうが画質は綺麗になる。従って、上記余りのビット数  $RN$  を  $Y$  のブロックにだけ分配する。上述の例では、 $YA = RN / (V \times H) = 180 / 100$  となる。そして、各ブロックに均等に分配した余りのビット数  $RN$  から  $Y$  のブロックのみに分配するビット数 ( $YA \times V \times H$ ) を減算して上記  $Y$  のブロックのみに均等分配しても分配しきれなかった余りのビット数  $RN$  を求める ( $RN = RN - YA \times V \times H$ )。上述の例では、 $RN = 180 - 100$  となる。

- 10 【0031】上述したように  $Y$  のブロックのみに均等に分配しているが、それでもまだ余りが生ずる場合がある。そこで、上記  $Y$  の分配で分配しきれなかった余りのビット数  $RN$  を  $V$  方向のブロック数の 4 倍で割り、 $Y$ 、 $R-Y$ 、 $B-Y$  の一番左側のブロックに 2 : 1 : 1 の割合で分配するビット数  $RB$  を求める ( $RB = RN / (V \times 4)$ )。すなわち、通常は左上のブロックから横方向のブロックについて処理を行なうが、右下のブロックに来たとき余りが多くあってももはや適当な分配を行うことはできない。そこで、本実施例では一番左側のブロックに分配するようにする。このときも、 $Y$  の方に多くの符号を与えるため例えば  $Y$  は  $R-Y$ 、 $B-Y$  の 2 倍を与えるようにし、 $Y$ 、 $R-Y$ 、 $B-Y$  の一番左側のブロックに夫々 2 : 1 : 1 の割合で分配する。これを実現するため上記  $RN$  を ( $V \times 4$ ) で割るようにしている。上述の例で、 $V = 10$  とすると、 $RB = 80 / (10 \times 2 + 10 + 10) = 2$  となる。そして、 $Y$  のブロックで分配した余りのビット数  $RN$  から全ての左側のブロックに分配するビット数 ( $RB \times 4 \times 4$ ) を減算して上記一番左側のブロックに分配しても分配しきれなかった余りのビット数  $RN$  を求める ( $RN = RN - RB \times V \times 4$ )。

- 30 【0032】上述した  $Y$ 、 $R-Y$ 、 $B-Y$  の一番左側のブロックに 2 : 1 : 1 の割合で分配する分配によってもまだ余りが生ずる場合がある。そこで、分配しきれなかった余りのビット数  $RN$  を  $V$  方向のブロック数で割り、 $Y$  の一番左側のブロックに均等に分配するビット数  $YB$  を求める ( $YB = RN / V$ )。そして、 $Y$ 、 $R-Y$ 、 $B-Y$  の一番左側のブロックに分配した余りのビット数  $RN$  から  $Y$  の一番左側のブロックに分配するビット数 ( $YB \times V$ ) を減算して  $Y$  の一番左側のブロックに均等に分配してもまだ分配しきれなかった余りのビット数  $RN$  を求める ( $RN = RN - YB \times V$ )。上述した  $Y$  の一番左側のブロックに均等に分配してもまだ余りが生ずる場合、後述するステップ  $S15$  でその余りは  $Y$  の左上端のブロックに加えてしまうようにする。

- 40 【0033】以上によりステップ  $S11$  の処理を終え、ステップ  $S12$  に進む。ステップ  $S12$  では先ず縦方向について処理を行うために縦方向のブロック数  $j$  を 0 にし ( $j = 0$ )、ステップ  $S13$  で縦方向のブロック数  $j$  が縦方向の所定ブロック数  $V$  より小さいか ( $j < V$ ) か否かを判別する。縦方向のブロック数  $j$  が縦方向の所定

ブロック数V以上のときはステップS14で所定ブロック数V以上のときはステップS14で発生符号量予測値F[i][j][k]を求め、縦方向のブロック数jをインクリメント(j=j+1)してステップS13に戻る。ここで、上記予測発生符号量F[i][j][k]のうち[i]がバンク(bank)でありi=0のときがY、i=1のときがB-Y、R-Yである。また、[i]がV(垂直)、[k]がH(水平)方向である。この場合において、Yの予測発生符号量F[0][j][0]は、一度目で求めた予測発生符号量F[0][j][0]に、余り分として全ての左端のブロックに分配するビット数RBを2倍にした値とYの左端のブロックに分配するビット数を加算した値となる(F[0][j][0]=F[0][j][0]+RB×2+YB)。また、B-Yの予測発生符号量F[1][j][0]は、一度目で求めた予測発生符号量F[1][j][0]に、余り分として全ての左端のブロックに分配するビット数RBを加算した値となり(F[1][j][0]=F[1][j][0]+RB)、また、R-Yの予測発生符号量F[1][j][H/2]は、一度目で求めた予測発生符号量F[1][j][H/2]に、余り分として全ての左端のブロックに分配するビット数RBを加算した値となる(F[1][j][0]=F[1][j][H/2])。ここで、図9に示すようにB-Y、B-Yは2つ合わせてYとなるから上記R-Yの予測発生符号量の水平[k]は[H/2]で表される。このようにして、上記ステップS13及びS14のループで予測発生符号量F[i][j][k]が縦方向のブロック数Vとなるまで繰り返し算出され、余りが左端のブロックに全部割り振られることになる。縦方向のブロック数jが縦方向の所定ブロック数V以上になると縦方向の全てのブロックの処理が終了したと判断してステップS15で最終段階の余りRNを左端のブロックの予測発生符号量F[0][0][0]に加えて(F[0][0][0]=F[0][0][0]+RN)図8のステップS16に進む。

【0034】図8のフローでは、前述した余りを全体のブロックに均等に分配する処理と、Yのブロックのみに均等に分配する処理を行なう。すなわち、ステップS16で縦方向のブロック数jを0にし(j=0)、ステップS17で縦方向のブロック数jがVより小さいか(j<V)否かを判別する。縦方向のブロック数jが縦方向の所定ブロック数Vより小さいときはステップS18で横方向のブロック数kを0にし(k=0)、ステップS20で横方向のブロック数kが横方向の所定ブロック数Hより小さいか(k<H)否かを判別する。横方向のブロック数kが横方向の所定ブロック数H以上のときはステップS19で縦方向のブロック数jをインクリメント(j=j+1)してステップS17に戻り、横方向のブロック数KがHより小さいときはステップS21に

進む。

【0035】ステップS21では、前段の処理で既に算出されているYの予測発生符号量F[0][j][k]に、全てのブロックに分配するビット数RAとYのブロックに分配するビット数YAを加算しYの予測発生符号量F[0][j][k]を求めるとともに(F[0][j][k]=F[0][j][k]+RA+YA)、前段のステップで既に算出されているCの予測発生符号量F[1][j][k]に、全てのブロックに分配するビット数RAを加算しCのブロックに分配するビット数YAを加算してYの予測発生符号量F[1][j][k]を求め(F[1][j][k]=F[1][j][k]+RA+YA)、横方向のブロック数kをインクリメント(k=k+1)してステップS20に戻る。

【0036】上記各ステップを繰り返し処理し、ステップS17で縦方向のブロック数jがV以上になるとY、Cの全てのブロックについての予測発生符号量の算出が済んだと判断して本フローの処理を終了する。

【0037】以上説明したように、第1実施例の画像データ圧縮装置30は、DC成分によって発生したDC符号量を記憶するDC符号量メモリ35と、AC成分によって発生したAC符号量を記憶するAC符号量メモリ36と、全てのブロックのDC成分によって発生した符号量の総和を記憶するDC総発生符号量記憶メモリ37と、Y用、C用夫々2種類の量子化テーブルを用いて量子化したブロックのAC成分によって発生した符号量の総和を記憶するACテーブル別総発生符号量記憶メモリ38、39と、量子化装置33から出力されたデータに対してDC符号量メモリ35及びAC符号量メモリ36を用いてハフマン符号化を施し、全てのブロックのDC成分によって発生した符号量の総和をDC総発生符号量記憶メモリ37に記憶するとともに、市松状に切換えられた2種類の量子化テーブル51~54毎のAC成分によって発生した符号量の総和をテーブル別AC総発生符号量記憶メモリ38、39に記憶するハフマン符号化器34とを設け、一度目の圧縮時に2つの量子化テーブル51と53、52と54を用いることによって量子化テーブルの変更による発生符号量の変化の予測を行ない目標とする発生する符号量に最適な量子化テーブルを作成し、二度目の圧縮によって目標とする符号量に圧縮するようにしているので、従来3度の“DCT→量子化→符号化”シーケンスにより符号化制御を行っていたものが、二度の動作で済むようになり処理の大幅な高速化を図ることができる。このように、一度の量子化でY、Cそれぞれ複数種の量子化テーブルを用いるようにしているので、動作シーケンスを1回減らすことによって高速処理が実現でき、画像圧縮の符号量制御に利用して好適である。

【0038】なお、本実施例では複数種の量子化テーブルとして、各ブロックのうちの市松模様状に分けた2種

の量子化テーブルを用いているが、複数種の量子化テーブルを一度の量子化で使用するものであればどのような数、種類の量子化テーブルであってもよいことは勿論である。

【0039】また、量子化テーブルの作成方法やその個数、テーブルの値等は前述した実施例に限られるものではなく任意のものが使用可能である。

#### 【0040】第2実施例

図10～図12は本発明に係る画像データ圧縮装置の第2実施例を示す図であり、本実施例は複数種の量子化テーブルを用いる方法として、1つのフレームを複数のフィールド（ライン）に分割して複数の量子化テーブルを使用するようにした例である。本実施例の説明にあたり第1実施例と同一構成部分には同一符号を付して重複部分の説明を省略する。まず、構成を説明する。図10は画像データ圧縮装置60のブロック図であり、この図において、画像データ圧縮装置60は、原画像データを記憶するフレームメモリ31から後述する2つの読出しモード（Frame Mode）（Field Mode）に従って読出した画像データをDCT演算するDCT装置61と、DCT演算の出力データを2種の量子化テーブルをフィールド毎に切換えて量子化演算する量子化装置62と、後述する各メモリ35～37、64、65を用いて第1実施例と同様の2つの動作モードmode1、mode2に従って量子化装置62の出力に対してハフマン符号化を行なうハフマン符号化器63と、DC成分によって発生したDC符号量を記憶するDC符号量メモリ35と、AC成分によって発生したAC符号量を記憶するAC符号量メモリ36と、全てのブロックのDC成分によって発生した符号量の総和を記憶するDC総発生符号量記憶メモリ37と、odd Field（奇数フィールド）のAC成分によって発生した符号量の総和を記憶するフィールド別総発生符号量記憶メモリ64と、even Field（偶数フィールド）のAC成分によって発生した符号量の総和を記憶するフィールド別総発生符号量記憶メモリ65と、これら全体の制御及び最適と思われる量子化テーブル及び、その時の各ブロックの発生符号量の予測値の算出を行なうCPU66とにより構成されている。

【0041】上記DCT装置61は、フレームメモリ31から画像データを読み込んでDCT演算を行ない、その演算結果量子化装置62に出力するもので、フレームメモリ31からのデータの読込には、以下の2つのモードを持つ。

（Frame Mode） 順番に一つのフレームとして、上のラインから読み込む。

（Field Mode） 先に奇数番目のラインだけを読込み続いて偶数番目のラインを読み込む。

【0042】上記ハフマン符号化器63は、量子化装置62から出力されたデータに対してDC符号量メモリ35及びAC符号量メモリ36を用いてハフマン符号化を

施し、全てのブロックのDC成分によって発生した符号量の総和をDC総発生符号量記憶メモリ37に記憶するとともに、所定のフィールド毎に切換えられた後述する2種類の量子化テーブル毎のAC成分によって発生した符号量の総和をフィールド別AC総発生符号量記憶メモリ64、65に記憶する。

【0043】また、上記フィールド別AC総発生符号量記憶メモリ64、65のうち、フィールド別AC総発生符号量記憶メモリ64には、オッドフィールドのAC成分によって発生した符号量の総和が、またフィールド別総発生符号量記憶メモリ65にはイーブンフィールドのAC成分によって発生した符号量の総和が記憶保持される。また、（Frame Mode）時には、フィールド別総発生符号量記憶メモリ64にのみ値が書き込まれ、フィールド別総発生符号量記憶メモリ65の値は意味を持たない。

【0044】図11は上記量子化装置63のブロック図である。図11において、量子化装置62は、オッドフィールドのY用量子化テーブルを記憶する量子化テーブルRAM71と、オッドフィールドのY用量子化テーブルを記憶する量子化テーブルRAM73と、イーブンフィールドのC用量子化テーブルを記憶する量子化テーブルRAM72と、イーブンフィールドのC用量子化テーブルを記憶する量子化テーブルRAM74と、DCT演算の結果と上記各量子化テーブルの値を乗算する（逆数を掛けて除算する）ことによって重み付けを変える量子化を行なう乗算器75とにより構成されている。

【0045】上記量子化テーブルRAM71～74は、以下のようにコンポーネント及びフィールドに対して切換えて用いられる。

（Frame Mode） Yに対して、量子化テーブルRAM71の量子化テーブルを、Cに対して量子化テーブルRAM72の量子化テーブルを用いる。

（Field Mode） Yのオッドフィールドに量子化テーブルRAM71の量子化テーブルを、イーブンフィールドに量子化テーブルRAM73の量子化テーブルを、Cのオッドフィールドに量子化テーブルRAM72を、イーブンフィールドに量子化テーブルRAM74の量子化テーブルを夫々用いる。

【0046】なお、テーブル1として用いられるY用量子化テーブル1及びC用量子化テーブル1と、テーブル2として用いられるY用量子化テーブル2及びC用量子化テーブル2は、例えば前記図5に示すような量子化テーブル51～54の値のものが使用される。

【0047】次に、本実施例の動作を説明する。

#### 全体動作

まず、フレームメモリ31には、原画像のデータが保持されているものとする。DCT演算部612はフレームメモリ31から画像データを読み込んでDCT演算を行ない、そのDCT演算出力は、量子化装置62に送られ

る。また、量子化装置62では、DCT演算装置61からの入力に対して量子化テーブルRAM71~74の値に従って量子化演算を行ない、演算結果をハフマン符号化器63に出力する。ハフマン符号化器63は、量子化されたデータに対してDC符号量メモリ35及びAC符号量メモリ36を用いてハフマン符号化を施し、全てのブロックのDC成分によって発生した符号量の総和をDC総発生符号量記憶メモリ37に記憶するとともに、1つのフレームを2つのフィールドに分割した各々の量子化テーブル毎のAC成分によって発生した符号量の総和をフィールド別AC総発生符号量記憶メモリ64、65に書込む。この場合、2つのフィールドに分割したうちのオッドフィールドのAC成分によって発生した発生符号量の総和はフィールドテーブル別総発生符号量記憶メモリ64に書込まれ、イーブンフィールドのAC成分によって発生した発生符号量の総和はフィールド別総発生符号量記憶メモリ65に書込まれる。

【0048】また、CPU66は上記各装置を以下のように動作させ、以下に述べる最適と思われる量子化テーブル及びその時の各ブロックの発生符号量の予測値の算出を行なう。

i) 先ず、目標とする総発生符号量に合わせて量子化テーブルを作成し、量子化テーブルRAM71、72、73、74にその逆数を書込む。

i i) 次いで、DCT (Field mode) →量子化 (Field mode) →ハフマン符号化 (mode 1) を行なう。

i i i) 後述のアルゴリズムに従って、量子化テーブル及びその時の各ブロックの発生符号量の予測を行ない、その値を量子化テーブルRAM71、72の量子化テーブル、DC符号量メモリ35及びAC符号量メモリ36に書込む。

i v) 次いで、DCT (Frame Mode) →量子化 (Frame Mode) →ハフマン符号化 (mode 2) を行なう。

#### 【0049】発生符号量予測アルゴリズム

先ず、量子化テーブル作成アルゴリズムについて述べる。

①事前に発生符号量がリニアとなるような数種類の量子化テーブルの作成方法を決めて番号付けをしておく。例えば、0~255の256通りとし、0から255へ増加するものとする。

②一度目のField Modeで用いる量子化テーブルに目標とする発生符号量 (CDN) から適当なスケーリングを2つのスケーリング値を選択して書込み、圧縮を行なう。この選択した2つのスケーリング値をS<sub>b</sub>、S<sub>h</sub>とする。

③DC総発生符号量記憶メモリ37のDC総発生符号量の値をDC<sub>all</sub>とし、スケーリング値S<sub>b</sub>、S<sub>h</sub>に対応するフィールド別総発生符号量記憶メモリ64、65のフィールド別総発生符号量の値をそれぞれAC<sub>s b</sub>、AC<sub>s h</sub>とすると、求める量子化テーブルの番号は例え

ば前記(式1)によって得ることができる。

【0050】次に、各ブロックの発生符号予測アルゴリズムについて述べる。

①目標とする発生符号量から、一度で発生したDC成分の符号量を減じた値と、各テーブルによって発生したAC成分の総和 (AC<sub>s b</sub>、AC<sub>s h</sub>) との比率を求める。

②図12 (a) に示す原画を同図 (b) に示すようにオッドフィールド (odd Field) とイーブンフィールド (even Field) の2つのフィールドに分割し、オッドフィールドとイーブンフィールドの同じブロックに対して各コンポーネントに対応するEOB長を減じ、EOB長を減算した値に上記①で求めた比率をそれぞれに乗じた値を求めて加える。

③次いで、上記②で加算した値を2分割し、図12

(c) に示すような上下2つのブロックのAC成分として割振る。

④上記③の値に、事前に減じておいたDC長の平均、及びEOB長を加えて、該当するブロックの予測発生符号量とする (図12 (d) 参照)。

【0051】以上説明したように、第2実施例の画像データ圧縮装置60は、AC成分によって発生したAC符号量の総和を記憶するフィールド別総発生符号量記憶メモリ64と、AC成分によって発生した符号量の総和を記憶するフィールド別総発生符号量記憶メモリ65を備え、一度目の圧縮時には一度目の圧縮時にはフレームをオッドとイーブンの2つのフィールドに分割し、各々に別の量子化テーブルを使用してその発生符号量に基づいて量子化テーブルの変更による全体及び各ブロックの発生符号量の予測を行ない、二度目の圧縮時にはフレームのデータとして目標とする符号量に圧縮するようにしているので、第1実施例の場合と同様に2度の圧縮で正確な符号量制御を行うことができるようになり、処理の大幅な高速化を図ることができる。特にフィールドモードが設けられている電子カメラに本画像データ圧縮装置60を適用して好適である。

【0052】なお、上記各実施例では画像データ圧縮装置をJPEGアルゴリズムに基づく画像データ圧縮装置に適用した例であるが、勿論これには限定されず、発生符号量の制御を行なうものであれば全ての装置に適用可能であることは言うまでもない。また、量子化テーブルを用いるものであれば何でもよく、その作成の仕方本実施例に限定されない。

【0053】また、上記各実施例は符号化部にハフマン符号化方法を用いているが、これに限らず、例えばハフマン符号のようなブロック符号に代えてエントロピー符号化として算術符号を用いるようにしてもよく、算術符号の一種であるQ-Coderを用いてもよい。

【0054】また、上記各実施例では、変換符号化方式にDCTを適用しているが、このDCT方式には限定さ

れず、例えば、アダマール変換、ハール (Harr) 変換、傾斜変換 (スラント変換)、対称性サイン変換などを用いた画像データ圧縮装置に適用することができる。

【0055】さらに、上記画像データ圧縮装置30、60や量子化装置33、62等を構成する回路や部材の数、種類などは前述した実施例に限られないことは言うまでもない。

【0056】

【発明の効果】請求項1、2、3及び4記載の発明によれば、一度目の圧縮時に前記複数種の量子化テーブルを用いるようにしているので、従来3度の“DCT→量子化→符号化”という動作によって符号化制御を行っていたものを2度の動作で済ますことができ、高速に処理を行なうことができ、画像データの圧縮に利用して好適である。

【0057】請求項5及び請求項6記載の発明によれば、一度目の圧縮時には、1つのフレーム (原画) を2つのフィールドに分割し、各々異なった量子化テーブルを用いて圧縮するようにしているので、2度の圧縮で正確な符号量制御を行なうことができ、フィールドモード

【図面の簡単な説明】

【図1】第1実施例に係る画像データ圧縮装置のブロック構成図である。

【図2】第1実施例に係る画像データ圧縮装置の量子化装置のブロック構成図である。

【図3】第1実施例に係る画像データ圧縮装置のY用量子化テーブルを示す図である。

【図4】第1実施例に係る画像データ圧縮装置のR-Y、B-Y用テーブルを示す図である。

【図5】第1実施例に係る画像データ圧縮装置の量子化テーブルの一例である。

【図6】第1実施例に係る画像データ圧縮装置の発生符号量の予測処理を示すフローチャートである。

【図7】第1実施例に係る画像データ圧縮装置の発生符

号量の子測処理を示すフローチャートである。

【図8】第1実施例に係る画像データ圧縮装置の発生符号量の予測処理を示すフローチャートである。

【図9】第1実施例に係る画像データ圧縮装置のY、B-Y、R-Yのブロックの構造を示す図である。

【図10】第2実施例に係る画像データ圧縮装置のブロック構成図である。

【図11】第2実施例に係る画像データ圧縮装置の量子化装置のブロック構成図である。

【図12】第2実施例に係る画像データ圧縮装置の発生符号量予測アルゴリズムを説明するための図である。

【図13】従来の画像データ圧縮装置のブロック構成図である。

【図14】従来の画像データ圧縮装置の量子化装置のブロック構成図である。

【図15】従来の画像データ圧縮装置のY用量子化テーブルを示す図である。

【図16】従来の画像データ圧縮装置のR-Y、B-Y用量子化テーブルを示す図である。

【符号の説明】

30、60 画像データ圧縮装置

31 フレームメモリ

32、61 DCT演算装置

33、62 量子化装置

34、63 ハフマン符号化器

35 DC符号量メモリ

36 AC符号量メモリ

37 DC総発生符号量記憶メモリ

38、39 ACテーブル別総発生符号量記憶メモリ

40、66 CPU

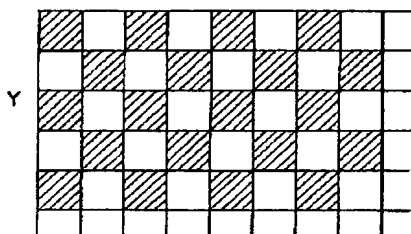
41~44、71~74 量子化テーブルRAM

45、75 乗算器

51~54 量子化テーブル

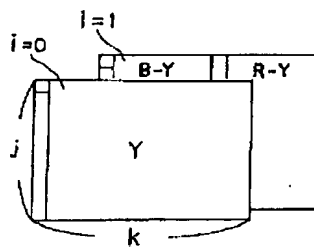
64、65 フィールド別総発生符号量記憶メモリ

【図3】



□ … 量子化テーブル51  
 ▨ … 量子化テーブル53

【図9】



【図15】

16,	11,	10,	16,	24,	40,	51,	61
12,	12,	14,	19,	26,	58,	60,	55
14,	13,	16,	24,	40,	57,	69,	56
14,	17,	22,	29,	51,	67,	80,	62
18,	22,	37,	58,	68,	109,	103,	77
24,	35,	55,	64,	81,	104,	113,	92
49,	64,	78,	87,	103,	121,	120,	101
72,	92,	95,	98,	112,	100,	103,	99

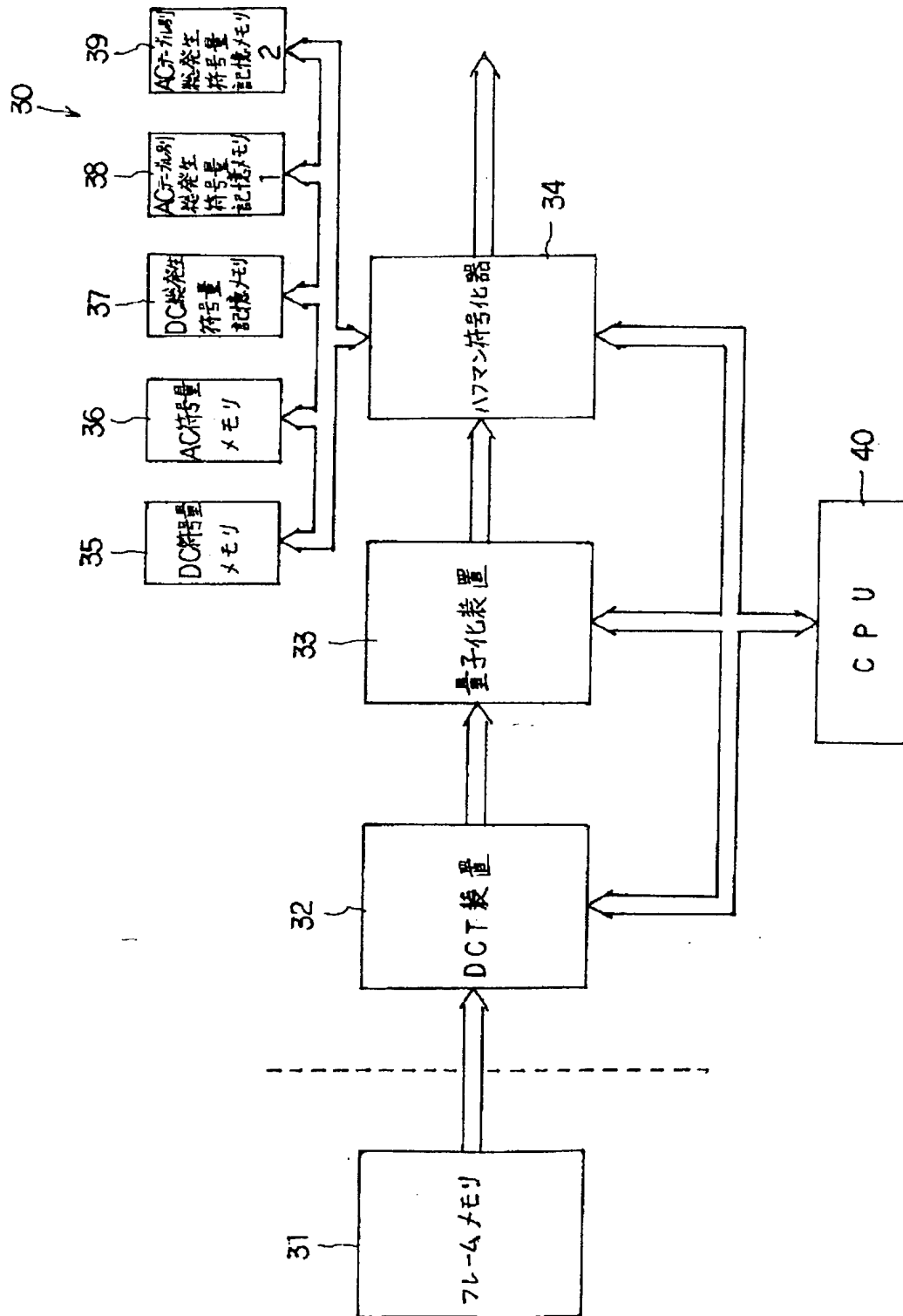
Y用  
量子化テーブル

【図16】

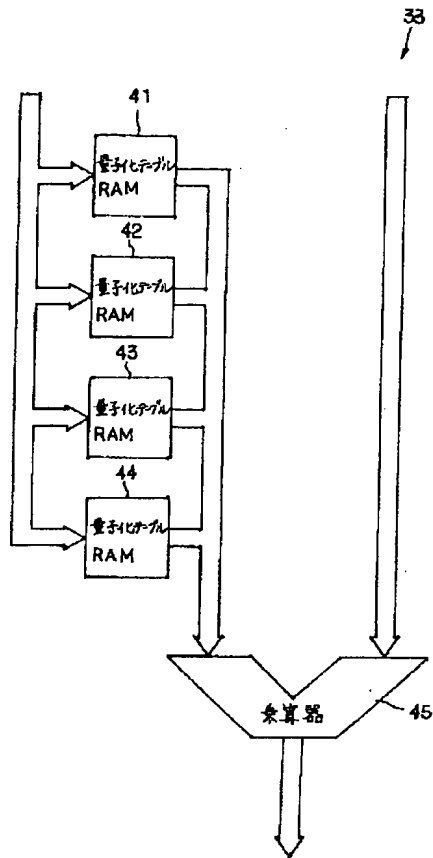
17,	18,	24,	47,	97,	97,	98,	98
18,	21,	26,	66,	97,	98,	98,	99
24,	26,	56,	97,	98,	98,	99,	99
47,	66,	97,	98,	98,	99,	99,	99
66,	97,	98,	98,	99,	99,	99,	99
97,	98,	98,	99,	99,	99,	99,	99
98,	98,	99,	99,	99,	99,	99,	99
98,	99,	99,	99,	99,	99,	99,	99

C用  
量子化テーブル

【図1】



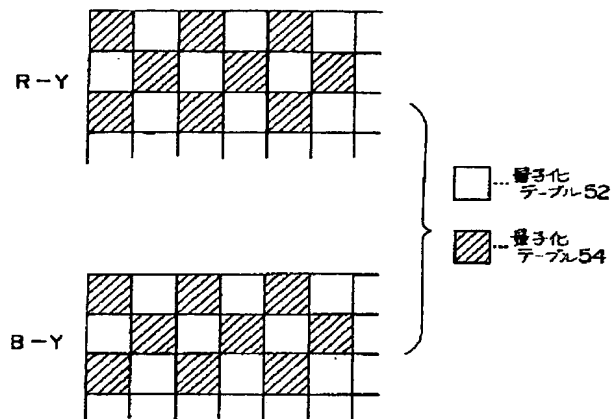
【図2】



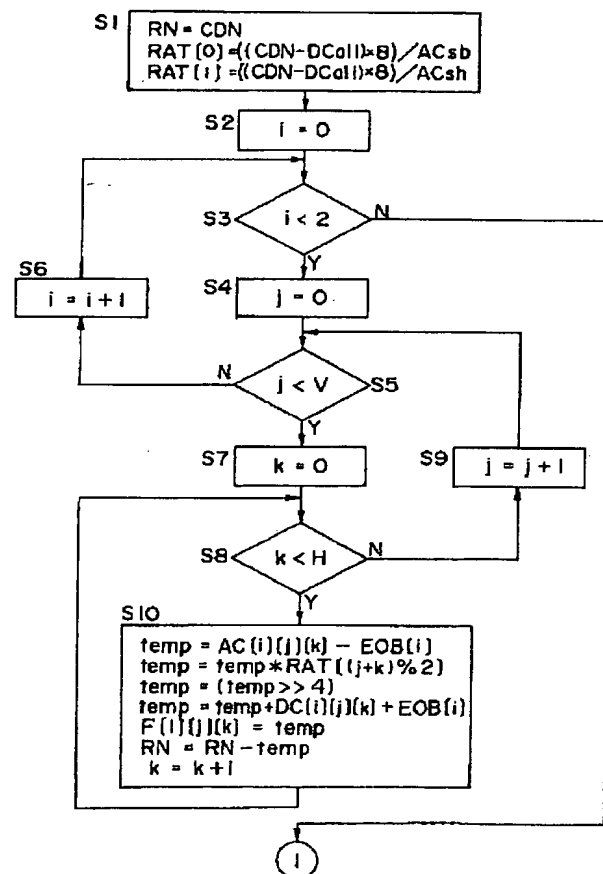
【図5】

4	21	21	35	52	87	111	133	
23	26	30	41	56	126	131	120	
30	28	35	52	87	124	150	122	
30	37	48	63	111	190	175	135	
39	48	80	126	148	238	225	168	
52	76	120	140	177	227	247	201	
107	140	170	190	225	255	255	220	
157	201	207	214	245	218	225	216	
4	34	52	102	216	216	216	216	
34	45	56	144	216	216	216	216	
52	56	122	216	216	216	216	216	
102	144	216	216	216	216	216	216	
216	216	216	216	216	216	216	216	
216	216	216	216	216	216	216	216	
216	216	216	216	216	216	216	216	
216	216	216	216	216	216	216	216	
216	216	216	216	216	216	216	216	
4	3	3	4	7	12	17	20	
3	3	4	5	7	19	20	18	
4	4	4	7	13	19	23	19	
4	5	6	10	17	29	27	21	
5	6	11	19	23	37	34	26	
7	10	18	21	27	35	38	31	
14	21	26	29	34	41	40	34	
24	31	32	33	37	33	34	33	
4	5	7	14	29	29	33	33	
5	6	7	19	29	33	33	33	
7	7	16	29	33	33	33	33	
14	19	29	33	33	33	33	33	
29	29	29	33	33	33	33	33	
29	29	33	33	33	33	33	33	
29	33	33	33	33	33	33	33	
33	33	33	33	33	33	33	33	

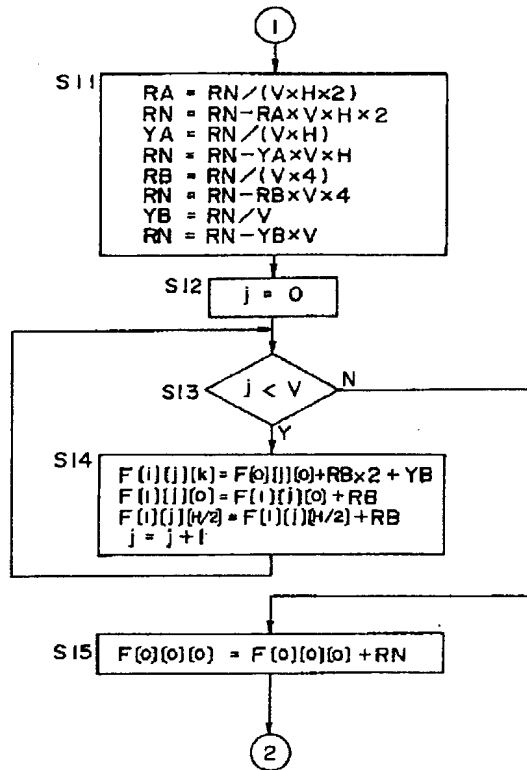
【図4】



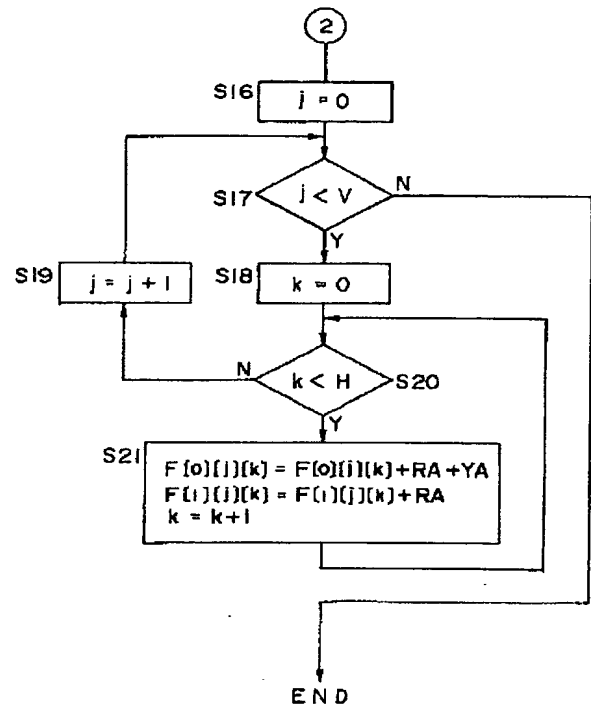
【図6】



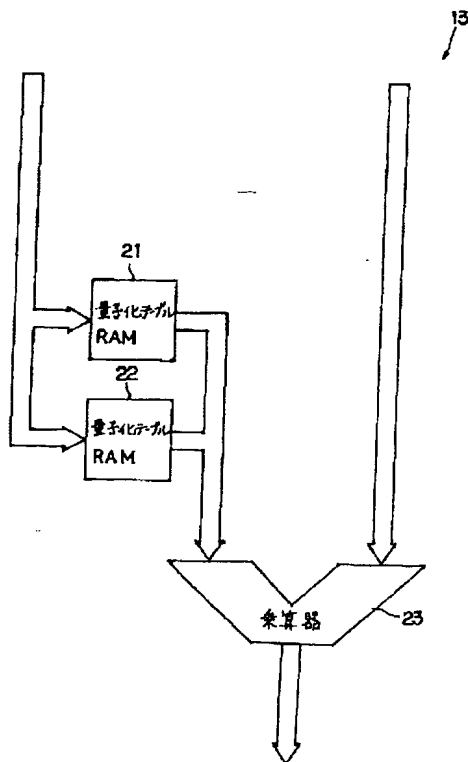
【図7】



【図8】

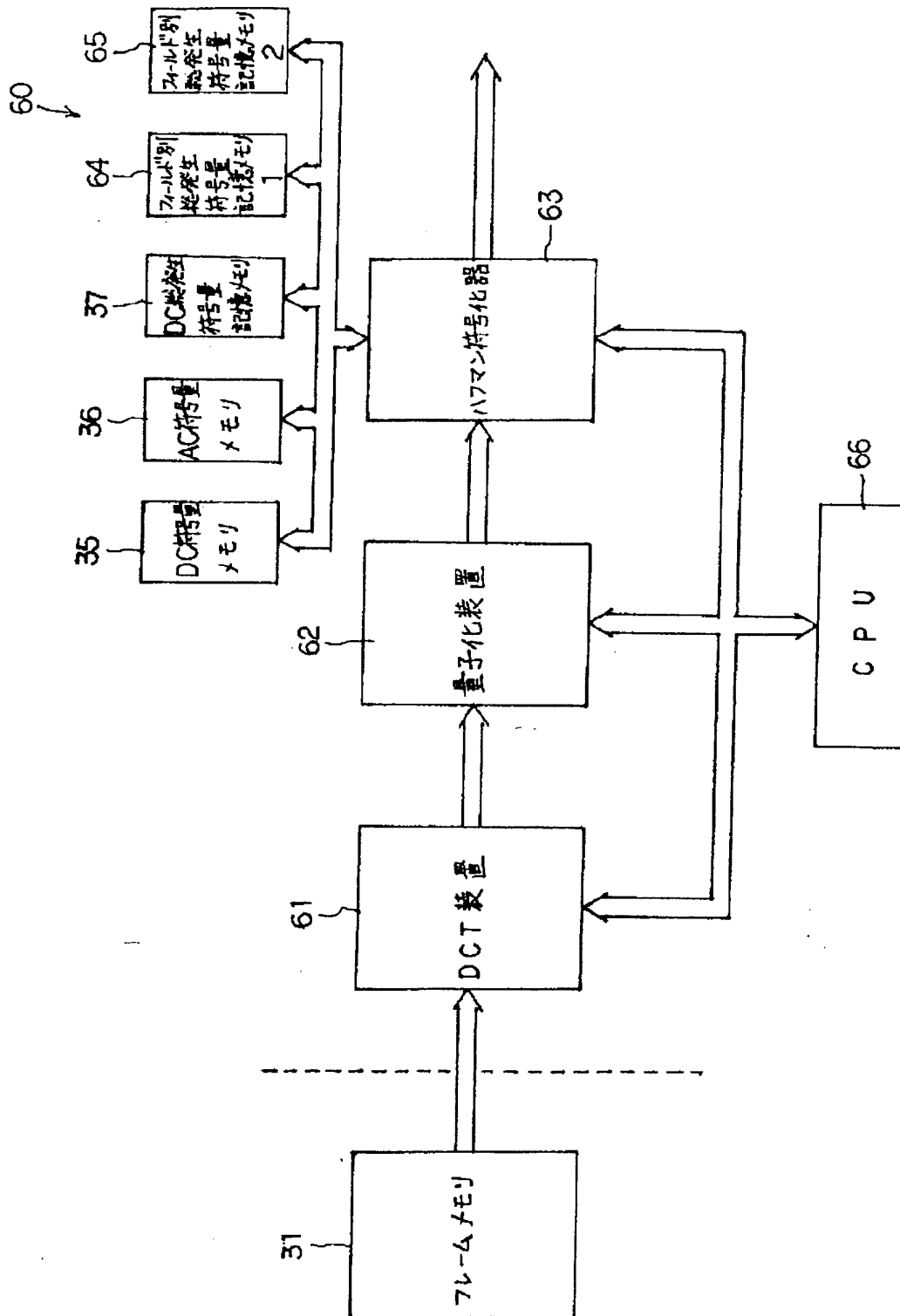


【図14】

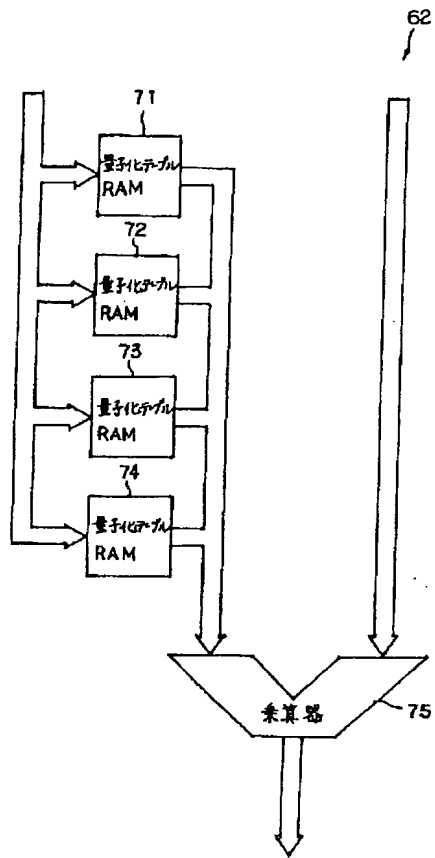




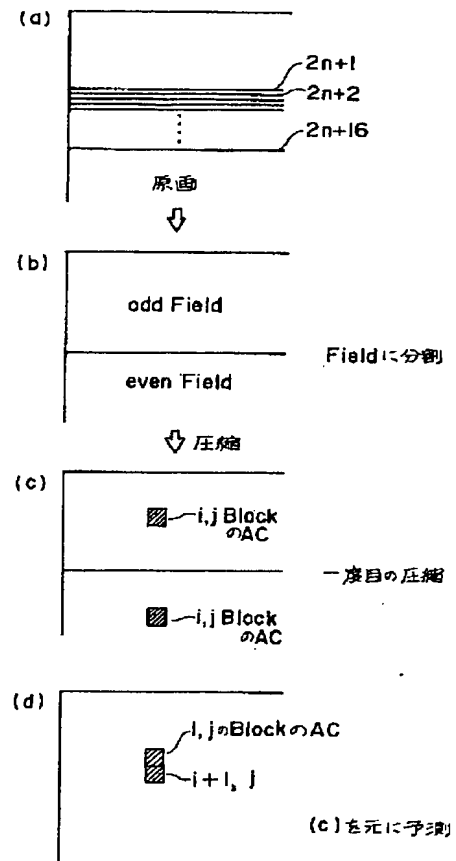
【図10】



【図11】



【図12】



【図13】

